Sim03 - Grading Rubric and Assessment Form

PROGRAMMER Secret Number: _____ GRADER Secret Number: _____

Grading annotation is **required** where lines are provided

Quality development and building process – must be easy to read and build

_____ / 5. Program compiles without warnings or errors; makefile is correctly configured
so that typing **`make`** or **`make -f <filename>`** will implement the building process
- all required code, libraries, etc. complete and available
- no more than 2 points credit if unecessary files such as unused library files, program configuration and/or meta-data files, etc. are included
- incorrect file name and/or format, or does not use required make file specifications: -10

_____
_____

_____ / 40. Program source code is easily readable and understandable (with significant consideration for the Programming Standards document)
≤ 5 pts. Difficult or impossible to read or understand, poor indenting and program structure
≤ 10 pts. Some parts difficult to read or have poor structure, but some program parts are clear
≤ 30 pts. Some parts difficult to read or have poor structure, but overall program process is clear
≤ 40 pts. Program is written and structured clearly, all parts are quickly and easily undersood
-1 pt for each single-letter variable or missing curly brace pairs at if/else/while/for/etc.
-5 pt for each global variable

_____
_____
_____
_____

Quality program development – specified items must be easy to identify and understand (no credit if data is not stored in an appropriate data structure, or any other assignment specifications not followed). **Grader must read and analyze this code; these parts do not require running the program**

_____ / 10. Program and code are structured well; functions are appropriately used to support program modularity; code is efficient and is not repeated unnecessarily (i.e., very little or no duplicated code)

_____
_____
_____

_____ / 15. Correct and complete management of FCFS-N and SJF-N. Code analysis must show appropriate display (running program not required). No credit if data is stored to file at the same time as it is displayed to the monitor.

_____
_____
_____

_____ / 15. Correct and complete programming of memory management, which would provide an output equivalent to the provided examples (running program not required). Provides code that would handle allocation success or failure as well as access success or failure.

_____
_____
_____

Program operation – program is compiled and run, with specified testing conditions

_____ / 15. Driver program runs correctly and clearly demonstrates the appropriate simulator operations
Program runs correctly as specified here:
- with monitor display for memory allocation to succeed with FCFS-N (+3 pt)
- with monitor display for memory allocation to fail with FCFS-N (+3 pt)
- with monitor display for memory allocation to succeed with SJF-N (+3 pt)
- with monitor display for memory allocation to fail with SJF-N (+3 pt)
- with monitor & file output for memory allocation to succeed with FCFS (+3 pt)

- **no credit here** if 1) program cannot be compiled and/or does not run, 2) program does not display with real-time delays as specified, 3) any non-student-created timers (e.g., sleep, usleep, etc.) are used

_____
_____
_____

_____ Valgrind test: -1 pt (up to 10 points) for each identified memory block not freed ("definitely lost")

_____ Specification not met or constraint not honored: See Michael

_____
_____

_____ / 100. Assignment Subtotal

Instructor grading area on this page. No student writing here.
However, please make sure this page is attached to the rubric page(s).

_____ / 100. Assignment Subtotal, less late submission reduction if appropriate

_____ / 25. Grader Score

_____

_____

_____ / 125. Total Simulator 03 Program Grade